UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/966,458	09/28/2001	Hong Wang	42390P11705	5276

7590 07/19/2006

Blakely, Sokoloff, Taylor & Zafman
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, CA 90025-1030

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 07/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/966,458	WANG ET AL.	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3,5-21 and 23-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3,5-21 and 23-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is responsive to Applicant's submission filed on April 24, 2006. Claims 1, 3, 5-21 and 23-30 are pending.

Response to Arguments

2. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection, as set forth below with reference to Jacklin. Applicant's amendment necessitated the new ground(s) of rejection.

Claim Objections

3. Claim 1 is objected to because the claim does not consistently use the term "handler" or "handling" to refer to the routines "for the processing of captured profiles." For example, the claim recites a "handler routine of the plurality of handling routines" at line 17. The other claims appear to use "handler" rather than "handling." Appropriate correction is required.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 3, 5-21 and 23-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,134,710 to Levine et al. (art of record, "Levine") in view of U.S. Patent No. 5,838,969 to Jacklin et al. (now made of record, "Jacklin").

With respect to claim 1 (currently amended), Levine discloses an event monitoring component for dynamic optimization (see, for example, the abstract) comprising:

(a) an event monitor to selectively capture a plurality of profiles of one or more microarchitecture events occurring in the execution of an application by a microprocessor, the selection of the profiles to be captured and the one or more events to be monitored being based upon configuration information supplied by a software component (see, for example, column 7, line 59 to column 8, line 11, which shows a performance monitor for monitoring events during execution, and column 8, lines 30-35, which shows capturing profiles of the events, and column 8, lines 14-16, which shows that the monitor is controlled or configured by software);

(b) a profile buffer to store the plurality of captured profiles of the one or more microarchitecture events (see, for example, column 11, lines 34-48, which shows a buffer for storing the profiles of the events);

(c) an interface through which the software component provides the configuration information to direct the operation of the event monitor (see, for example, column 8, lines 16-22, which shows an interface for configuring the operation of the monitor); and

(d) one or more monitor control vectors, the monitor control vectors storing the configuration information provided by the software component (see, for example, FIGS. 3A and 3B, which show monitor control registers or vectors for storing the configuration).

Levine further discloses a handler routine to process the profiles of the events, and further discloses that the monitor control registers or vectors include fields to enable the handler routine (see, for example, interrupt handling routine 580 in FIG. 7 and column 8, lines 24-35). Levine does not expressly disclose the limitation wherein the software component includes a plurality of

handling routines for the processing of captured profiles, and the limitation wherein each monitor control vector includes a handler field to hold a pointer to a handler routine of the plurality of handling routines, the handler routine being selected by the software component to process the profiles of the microarchitecture event.

However, some reference to the address of the handler routine, or a pointer to the handler routine, is inherent to Levine. Moreover, Jacklin expressly discloses a handler routine (see, for example, event handlers 348 in FIG. 3), and a pointer to the handler routine held in a handler field of a monitor control vector (see, for example, pointer 676 in FIG. 6, and column 5, lines 50-63). Jacklin further discloses an application or software component that includes a plurality of such handler routines (see, for example, column 3, lines 40-46). The software component selects one or more handler routines to process selected events (see, for example, column 3, lines 47-56). This arrangement allows different handler routines to have different implementations for different events (see, for example, column 4, lines 10-21), and allows the software component to selectively configure how it handles events (see, for example, column 2, lines 18-31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate into Levine a plurality of handler routines and a handler field with which the software component selects one of the handler routines, as Jacklin suggests, so as to selectively configure different handler routines to process the profiles of different events.

With respect to claim 3 (previously presented), the rejection of claim 1 is incorporated, and Levine further discloses the limitation wherein each monitor control vector further includes:

(a) a control field to specify a microarchitecture event to monitor (see, for example, column 8, lines 44-52, which shows a control field for selecting events to monitor); and

(b) a trigger field to specify when the microarchitecture event is monitored (see, for example, column 8, lines 36-44, which shows a threshold field or trigger field for specifying when to count or monitor events).

With respect to claim 5 (previously presented), the rejection of claim 1 is incorporated, and Levine further discloses the limitation wherein the profile buffer comprises a first level buffer for initial storage of the captured profiles of the one or more microarchitecture events and a second level buffer for subsequent storage of the captured profiles of the one or more microarchitecture events (see, for example, column 10, line 63 to column 11, line 3, which shows storing the profile data in sampled instruction and data address registers, i.e. a first level buffer, and subsequently copying the data to tables in memory, i.e. a second level buffer).

With respect to claim 6 (original), the rejection of claim 5 is incorporated, and Levine further discloses the limitation wherein the first level buffer is a register file (see, for example, the registers of the first level buffer 530 and 540 in FIG. 7).

With respect to claim 7 (previously presented), the rejection of claim 5 is incorporated, and Levine further discloses the limitation wherein the second level buffer is a memory buffer that is architecturally visible to the software component (see, for example, column 11, lines 45-53, which shows that the second level buffer in memory is accessible by and thus architecturally visible to the software).

With respect to claim 8 (currently amended), the rejection of claim 1 is incorporated, and Levine further discloses the limitation wherein the captured event profiles of each monitored

microarchitecture event are made available to the handler routine selected by the software component for processing according to the handler field of the monitor control vector (see, for example, column 11, lines 37-42 and 53-56, which shows that the profile data of each event is made available for processing).

With respect to claim 9 (previously presented), the rejection of claim 1 is incorporated, and Levine further discloses the limitation wherein the event monitoring apparatus initiates an interrupt or special event handler to notify the software component when captured event profiles are available for processing (see, for example, column 10, line 63 to column 11, line 3, which shows signaling an interrupt when profile data is available).

With respect to claim 10 (currently amended), Levine discloses a microprocessor (see, for example, FIG. 1 and column 6, lines 57-61), comprising:

(a) an execution pipeline (see, for example, the execution units in FIG. 2 and the pipelined instruction cycle 14 in FIG. 4);

(b) one or more event monitor hardware components coupled to the execution pipeline to selectively monitor one or more microarchitecture events chosen by a software component in the execution of a program and to capture event profiles selected by a software component for the chosen microarchitecture events (see, for example, column 7, line 59 to column 8, line 11, which shows performance monitors for monitoring events during execution, and column 8, lines 30-35, which shows capturing profiles of the events, and column 8, lines 14-16, which shows that the monitors are controlled or configured by software);

(c) one or more monitor control vectors to store configuration information provided by a software component in connection with the operation of the one or more event monitor hardware components (see, for example, column 8, lines 14-22, which shows monitor control registers or vectors for controlling or configuring the monitors by software).

Levine further discloses a handler routine to process the profiles of the events, and further discloses that the monitor control registers or vectors include fields to enable the handler routine (see, for example, interrupt handling routine 580 in FIG. 7 and column 8, lines 24-35). Levine does not expressly disclose the limitation wherein each monitor control vector includes a handler field to contain a pointer to a handler routine selected by the software component for the microarchitecture event from a plurality of handler routines.

However, some reference to the address of the handler routine, or a pointer to the handler routine, is inherent to Levine. Moreover, Jacklin expressly discloses a handler routine (see, for example, event handlers 348 in FIG. 3), and a pointer to the handler routine held in a handler field of a monitor control vector (see, for example, pointer 676 in FIG. 6, and column 5, lines 50-63). Jacklin further discloses an application or software component that includes a plurality of such handler routines (see, for example, column 3, lines 40-46). The software component selects one or more handler routines to process selected events (see, for example, column 3, lines 47-56). This arrangement allows different handler routines to have different implementations for different events (see, for example, column 4, lines 10-21), and allows the software component to selectively configure how it handles events (see, for example, column 2, lines 18-31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate into Levine a plurality of handler routines and a handler field with

which the software component selects one of the handler routines, as Jacklin suggests, so as to selectively configure different handler routines to process the profiles of different events.

Levine further discloses:

(d) a profile buffer to store captured microarchitecture event profiles (see, for example, column 11, lines 34-48, which shows a buffer for storing profiles of the events).

With respect to claim 11 (previously presented), the rejection of claim 10 is incorporated, and Levine further discloses the limitation wherein the captured profiles of the one or more microarchitecture events stored in the buffer are made available to the handler routine selected by the software component for optimization processing (see, for example, column 11, lines 37-42 and 53-56, which shows that the profile data of each event is made available for processing, and column 12, lines 1-6, which shows that the processing is for optimization).

With respect to claim 12 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 13 (original), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

With respect to claim 14 (original), the limitations recited in the claim are analogous to those of claim 7 (see the rejection of claim 7 above).

With respect to claim 15 (original), the rejection of claim 14 is incorporated, and Levine further discloses the limitation wherein the first level buffer is comprised of a plurality of register frames and the second level buffer is comprised of a plurality of memory buffers, with one of the

frames of the first level buffer and one of the memory buffers in the second level buffer being assigned to each monitored microarchitecture event (see, for example, column 10, line 63 to column 11, line 3, which shows that the first level buffer is comprised of a plurality of registers and that the second level buffer is comprised of a plurality of tables in memory, and column 8, lines 24-35, which shows that the buffers are employed and thus assigned for each selected or monitored event).

With respect to claim 16 (original), the rejection of claim 15 is incorporated, and Levine further discloses the limitation wherein the profiles of a microarchitecture event stored in a frame assigned to the microarchitecture event in the first level buffer memory are spilled into a buffer assigned to the microarchitecture event in the second level memory buffer when the frame assigned to the microarchitecture event in the first level memory buffer is fully allocated or when a condition established by the software component is met (see, for example, column 10, line 63 to column 11, line 3, which shows copying or spilling the contents of the first level buffer to the second level buffer when an interrupt is serviced or met, and column 8, lines 24-35, which shows establishing the condition for the interrupt).

With respect to claim 17 (original), the rejection of claim 16 is incorporated, and Levine further discloses the limitation wherein the captured profiles of a microarchitecture event are made available to the handler routine when the buffer assigned to the event in the second level memory buffer is fully allocated or when a condition established by the software component is met (see, for example, column 11, lines 37-42 and 53-56, which shows that the profile data of

each event is made available for processing when an interrupt is serviced or met, and column 8, lines 24-35, which shows establishing the condition for the interrupt).

With respect to claim 18 (currently amended), the limitations recited in the claim are analogous to those of claim 3 (see the rejection of claim 3 above).

With respect to claim 19 (original), although Levine further discloses that the events are monitored during pipelined execution (see, for example, column 6, lines 4-12), Levine does not expressly disclose the limitation wherein the one or more microarchitecture events are monitored during an exception detection stage of the execution pipeline.

However, the system of Levine inherently detects exceptions. Exceptions, such as interrupts, must be detected in order to be handled (see, for example, interrupt handling routine 580 in FIG. 7). Any stage in the pipelined instruction cycle may comprise exception detection, such as the execute instruction stage (see, for example, FIG. 4).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to monitor events during a stage of the execution pipeline, as taught by Levine, such as during an exception detection stage.

With respect to claim 20 (original), although Levine further discloses that the events are monitored during pipelined execution (see, for example, column 6, lines 4-12) and that the profiles are stored in a buffer (see, for example, column 11, lines 34-48), Levine does not expressly disclose the limitation wherein the captured microarchitecture event profiles are stored in the memory buffer during a write back stage of the execution pipeline.

However, the system of Levine includes a load/store unit for reading and writing to memory (see, for example, FIG. 2). Writing back to memory may occur, for example, during the complete instruction stage of the pipelined instruction cycle (see, for example, FIG. 4).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to store the profiles in the memory buffer, as taught by Levine, during a stage of the execution pipeline that includes a write back stage, such as during the complete instruction stage.

With respect to claim 21 (currently amended), Levine discloses a method (see, for example, the abstract) comprising:

(a) receiving configuration information from a software component, the configuration information directing the capturing of certain profiles for the monitoring of one or more selected microarchitecture events connected with the operation of a microprocessor in executing an application (see, for example, column 8, lines 14-22, which shows software for controlling or configuring the monitoring of events), wherein receiving the configuration information from the software component comprises receiving information regarding the setting of one or more monitor control vectors (see, for example, FIGS. 3A and 3B, which show monitor control registers or vectors for storing the configuration).

Levine further discloses a handler routine to process the profiles of the events, and further discloses that the monitor control registers or vectors include fields to enable the handler routine (see, for example, interrupt handling routine 580 in FIG. 7 and column 8, lines 24-35). Levine does not expressly disclose the limitation wherein each monitor control vector includes a handler

field to contain a pointer to a handler routine selected by the software component for processing of captured profiles of the microarchitecture event.

However, some reference to the address of the handler routine, or a pointer to the handler routine, is inherent to Levine. Moreover, Jacklin expressly discloses a handler routine (see, for example, event handlers 348 in FIG. 3), and a pointer to the handler routine held in a handler field of a monitor control vector (see, for example, pointer 676 in FIG. 6, and column 5, lines 50-63). Jacklin further discloses an application or software component that includes a plurality of such handler routines (see, for example, column 3, lines 40-46). The software component selects one or more handler routines to process selected events (see, for example, column 3, lines 47-56). This arrangement allows different handler routines to have different implementations for different events (see, for example, column 4, lines 10-21), and allows the software component to selectively configure how it handles events (see, for example, column 2, lines 18-31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate into Levine a plurality of handler routines and a handler field with which the software component selects one of the handler routines, as Jacklin suggests, so as to selectively configure different handler routines to process the profiles of different events.

Levine further discloses:

(b) monitoring the one or more selected microarchitecture events and capturing the selected profiles of the one or more microarchitecture events (see, for example, column 7, line 59 to column 8, line 11, which shows performance monitors for monitoring the events during execution, and column 8, lines 30-35, which shows capturing profiles of the events);

(c) storing the captured event profiles in a profile buffer (see, for example, column 11, lines 34-48, which shows a buffer for storing profiles of the events); and

(d) making the profiles of the event available to the software component for optimization processing (see, for example, column 11, lines 37-42 and 53-56, which shows that the profile data of each event is made available for processing, and column 12, lines 1-6, which shows that the processing is for optimization).

With respect to claim 23 (previously presented), the limitations recited in the claim are analogous to those of claim 3 (see the rejection of claim 3 above).

With respect to claim 24 (original), the limitations recited in the claim are analogous to those of claim 5 (see the rejection of claim 5 above).

With respect to claim 25 (previously presented), the limitations recited in the claim are analogous to those of claim 6 (see the rejection of claim 6 above).

With respect to claim 26 (previously presented), the limitations recited in the claim are analogous to those of claim 7 (see the rejection of claim 7 above).

With respect to claim 27 (original), the rejection of claim 26 is incorporated, and Levine further discloses assigning a register in the first stage and a memory buffer in the second stage to each monitored microarchitecture event (see, for example, column 8, lines 24-35, which shows that the buffers are employed and thus assigned for each selected or monitored event).

With respect to claim 28 (original), the rejection of claim 27 is incorporated, and Levine further discloses storing the profiles of each monitored microarchitecture event in the register assigned to the microarchitecture event in the first stage as the profiles of the microarchitecture event are captured (see, for example, column 10, line 63 to column 11, line 3, which shows storing the profile data in the first stage when it is captured).

With respect to claim 29 (original), the rejection of claim 27 is incorporated, and Levine further discloses spilling the profiles of a microarchitecture event from the register assigned to the microarchitecture event in the first stage to the memory buffer assigned to the microarchitecture event in the second stage when the register is fully allocated or when a condition established by the software component is met (see, for example, column 10, line 63 to column 11, line 3, which shows copying or spilling the contents of the first stage to the second stage when an interrupt is serviced or met, and column 8, lines 24-35, which shows establishing the condition for the interrupt).

With respect to claim 30 (original), the rejection of claim 29 is incorporated, and Levine further discloses notifying the software component when the register assigned to the event in the second stage of the memory buffer is fully allocated or when a condition established by the software component is met (see, for example, column 10, line 63 to column 11, line 3, which shows signaling an interrupt to notify the software, and column 8, lines 24-35, which shows establishing the condition for the interrupt).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure. U.S. Patent No. 6,728,949 to Bryant et al. discloses a method and system for periodic trace sampling using a mask to qualify trace data. U.S. Patent No. 5,355,484 to Record et al. discloses dynamically established event monitors in event management services of a computer system.

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER